**Работа с коллекциями классов**

|  |  |
| --- | --- |
| Одна из самых замечательных идей (правил) в объектно-ориентированном программировании является абстрагирование. При соблюдении этого правила, мы отдаляемся от кода и работаем с объектами, их свойствами, методами, как будто они существуют в нашем, физическом мире. Главное, что отличает класс от процедуры – возможность одновременного запуска нескольких экземпляров класса. Например, на основе класса «Cat» (Кот) можно создать несколько объектов и задать различные значения свойствам «Name» (Имя) и «Color» (Окраска). Для простой демонстрации создадим класс Cat с перечисленными ранее свойствами: Код класса:  Option Explicit  Private NValue As String  Private CValue As String  Property Let Name(NameValue As String)  NValue = NameValue  End Property  Property Get Name() As String  Name = NValue  End Property  Property Let Color(ColorValue As String)  CValue = ColorValue  End Property  Property Get Color() As String  Color = CValue  End Property        Код вызова класса в модуле:  Sub TestCat()  Dim Kity As Cat  Set Kity = New Cat  Kity.Color = "Рыжий"  Kity.Name = "Барсик"  Debug.Print Kity.Color, Kity.Name  End Sub  В примере выше создан объект Kity на основе класса Cat. Свойствам объекта задали значения. Создадим массив объектов на основе класса Cat (стаю котов):  Sub ManyCats()  Dim Cats(1 To 3) As Cat’объявляем массив типа Cat  Dim i As Integer  For i = 1 To 3  Set Cats(i) = New Cat’создаем объект и присваиваем значению массива  Next i  Cats(1).Color = "Белый"’задаем значения свойства  Cats(2).Color = "Рыжий"  Cats(3).Color = "Серый"  For i = 1 To 3  Debug.Print Cats(i).Color’читаем свойство  Next i  End Sub        Никто не запрещает создать многомерный массив… Все отлично. Мы создали стаю котов из 3 штук, читаем и задаем их свойства по индексу массива. Но чтобы добавить еще одного кота, необходимо переобъявлять массив, а для удаления кота присвоить значение Nothing. Добавление объекта в массив будет выглядеть так:  Dim Cats() As Cat 'обратите внимание, объявляем массив без размера  Dim i As Integer  ReDim Cats(1 To 3) As Cat 'меняем размер первый раз  For i = 1 To 3  Set Cats(i) = New Cat 'заполняем  Next i  Cats(1).Color = "Рыжий" 'задаем свойства  Cats(2).Color = "Серый"  Cats(3).Color = "Белый"  ReDim Preserve Cats(1 To 4) As Cat 'меняем размер второй раз  Set Cats(4) = New Cat 'заполняем  Cats(4).Color = "Дымчатый" 'задаем свойства  For i = 1 To 4  Debug.Print Cats(i).Color 'читаем  Next i        А при удалении объекта из массива нужно быть осторожным:  Dim Cats(1 To 3) As Cat 'объявляем массив  Dim i As Integer  For i = 1 To 3  Set Cats(i) = New Cat 'заполняем  Next i  Cats(1).Color = "Рыжий" 'задаем свойства  Cats(2).Color = "Серый"  Cats(3).Color = "Белый"  Set Cats(2) = Nothing 'удаляем объект из массива  Debug.Print Cats(1).Color 'читаем по отдельности  ' Debug.Print Cats(2).Color потому что здесь будет ошибка. Объекта нет...  Debug.Print Cats(3).Color        В данном случае мы не изменяем размерность массива, просто присваиваем объекту значение Nothing. Если нужно пройти по всем объектам массива и прочитать их свойства, можно воспользоваться циклом For Each…Next.  Dim Cats(1 To 3) As Cat 'объявляем массив  Dim i As Integer  For i = 1 To 3  Set Cats(i) = New Cat 'заполняем  Next i  Cats(1).Color = "Рыжий" 'задаем свойства  Cats(2).Color = "Серый"  Cats(3).Color = "Белый"  Dim c As Variant 'объявляем переменную как вариант (как cat не пройдет)  For Each c In Cats 'проходим по массиву  Debug.Print c.Color 'читаем свойство  Next c        Однако не следует забывать, что переменная, с помощью которой проходим по массиву должна иметь тип Variant. Хороший способ использования нескольких одинаковых объектов, это объединить их в коллекцию. Коллекция это объект, построенный на основе класса Collection, предназначенный для хранения данных (в том числе и других объектов). Он очень похож на одномерный массив. Индекс коллекции всегда начинается с 1. Свойства объекта Collection: Count – содержит число элементов (Item) коллекции. Item(index) – возвращает значение элемента (объект) по его индексу. Свойство также может вернуть значение по ключу. Об этом ниже… Методы: Add(Item, [Key], [Before], [After]) – Добавляет элемент в коллекцию. Параметр Item должен содержать добавляемое значение или объект. Это обязательный параметр. Остальные параметры не являются обязательными: Key – строковый ключ для элемента. Он должен быть уникальным. After и Before – выражение, указывающее перед каким или после какого элемента вставить новый. Если параметры числовые, они могут принимать значения от 1 до Count и вставка пройдет по индексу элементов, если параметры строковые, они должны содержать существующие названия ключей, и вставка будет происходить по ключам. Нельзя указывать два этих параметра сразу, только по отдельности. Ну как тут не вспомнить о коллекции Sheets… Вот пример создания коллекции и добавление элементов:  Dim i As Integer  Dim Cats As Collection 'объявляем коллекцию  Set Cats = New Collection 'создаем коллекцию  For i = 1 To 3  Cats.Add Item:=New Cat, Key:="Cat\_" & i 'добавляем в коллекцию объекты  Next i    Cats(1).Color = "Рыжий" 'задаем свойства  Cats(2).Color = "Белый"  Cats(3).Color = "Черный"    Debug.Print Cats.Item("Cat\_1").Color 'читаем свойства  Debug.Print Cats.Item(2).Color  Debug.Print Cats("Cat\_1").Color    Dim c As Cat 'еще способ читать свойства  For Each c In Cats  Debug.Print c.Color  Next c        Очень интересная вещь: при использовании конструкции For Each…Next, переменная c объявлена как Cat, а это означает ,что при написании кода, нам доступен просмотр свойств и методов класса cat. Это удобно. Вот пример удаления объекта из коллекции:  Dim i As Integer  Dim Cats As Collection 'объявляем коллекцию  Set Cats = New Collection 'создаем коллекцию  For i = 1 To 3  Cats.Add Item:=New Cat, Key:="Cat\_" & i 'добавляем в коллекцию объекты  Next i  Cats(1).Color = "Рыжий" 'задаем свойства  Cats(2).Color = "Белый"  Cats(3).Color = "Черный"  Cats.Remove 2 'Удаляем второй объект из коллекции  ' Cats.Remove ("Cat\_2") 'по ключу удалять так же можно  Dim c As Cat 'еще способ читать свойства  For Each c In Cats  Debug.Print c.Color  Next c  Debug.Print Cats("Cat\_3").Color 'число объектов изменилось и обращаться к ним лучше по ключам        При удалении объектов из коллекции меняется свойство Count. В работе с коллекциями мне больше всего нравиться использование ключей.  Dim i As Integer  Dim c As Cat  Dim Cats As Collection 'объявляем коллекцию  Set Cats = New Collection 'создаем коллекцию  Set c = New Cat  c.Name = "Барсик"  Cats.Add Item:=c, Key:=c.Name 'добавляем в коллекцию объекты  Set c = New Cat  c.Name = "Дымок"  Cats.Add Item:=c, Key:=c.Name  Set c = New Cat  c.Name = "Ушлепок"  Cats.Add Item:=c, Key:=c.Name  Cats("Дымок").Color = "Дымчатый" 'обращаемся к элементу коллекции по ключу  Debug.Print Cats("Дымок").Color        То есть даже, если число элементов изменится, всегда можно достучаться до нужного элемента по его ключу. Создание класса со свойством, содержащим коллекцию В файле[csvDataLoader.zip](http://excelworld.ru/Author/ratboy/ClassModule/csvDataLoader.zip) предложено практическое применение вышеописанной теории с котиками. Только вместо котиков ячейки. Некоторые программы создают в результате своей деятельности файлы \*.csv, которые содержат строки, в которых данные разделены запятыми. Я не нашел способа доступа к данным по номеру строки и столбца в vba, пришлось сделать самому. В примере создано подобие объектной модели для таких файлов, то есть адаптер для доступа к данным. Можно получать доступ к данным по номеру строки и столбца. Более того, в объектную модель можно загрузить несколько файлов и доступ будет осуществляться по имени файла (индексу), номеру строки и столбца. То есть получается фактически аналог книги Excel. Конечно, пример только учебный, в нем нет обработки ошибок, проверки многих исключений, не реализован поиск. Хорошо бы реализовать запись данных по индексам. Но пример этот хорошо демонстрирует способы работы с коллекциями и массивами объектов, построенных на основе собственных классов. Так же демонстрирует способ разработки внятной объектной модели, которая не реализована в Excel изначально. Подобные адаптеры для доступа к данным можно создать для любых типов файлов, в том числе и для пользовательских бинарных форматов, особенно, если планируется часто работать с ними. Кстати вот поиск для нашего адаптера:  Public Function SearchFirstCell(Data As String, Optional Column As Long = 0) As Cell  Dim r As Long  Dim c As Long  Dim cl As Cell  Select Case Column  Case 0  For r = 1 To UBound(DataArray, 1)  For c = 1 To UBound(DataArray, 2)  If DataArray(r, c).Value = Data Then  Set SearchFirstCell = DataArray(r, c)  Exit Function  End If  Next c  Next r  Case Else  For r = 1 To UBound(DataArray, 1)  If DataArray(r, Column).Value = Data Then  Set SearchFirstCell = DataArray(r, Column)  Exit Function  End If  Next r  End Select  Set cl = New Cell  cl.Value = "Указанное значение не найдено"  Set SearchFirstCell = cl  End Function        Метод осуществляет точный поиск первой ячейки. Требует указания искомого значения. Параметр столбец является необязательным. При его указании поиск осуществляется только по указанному столбцу. Метод возвращает объект. Пример использования:  Debug.print DataReader. SearchFirstCell("202”).Row        Найдет ячейку, содержащую значение 202 и вернет ее номер строки.  Желаю успехов в разработке ratboy [ondister@gmail.com](mailto:ondister@gmail.com) | |
|  |  |
| * [1](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113) * [2](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113) * [3](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113) * [4](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113) * [5](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113)   Категория: [Модули классов](http://www.excelworld.ru/publ/vba/class_module/42) | Добавил: [ratboy](javascript://) (11.04.2012) [E](javascript://) | |
|  |  |
| Просмотров: **8049** | Комментарии: **4** | Рейтинг: **5.0**/**7** | |

|  |  |
| --- | --- |
| Всего комментариев: **4** |  |
| Порядок вывода комментариев:         |  | | --- | | **0**  http://s5.ucoz.net/img/icon/thumbu_.png http://s5.ucoz.net/img/icon/thumbd_.png  **[1](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113" \l "ent67)**    **Пытливый**   (21.04.2012 13:14)     Файл не качаеится |  |  | | --- | | [Спам](javascript://)**0**  http://s5.ucoz.net/img/icon/thumbu_.png http://s5.ucoz.net/img/icon/thumbd_.png  **[2](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113" \l "ent68)**    **Пытливый**   (21.04.2012 13:15)     csvDataLoader.zip  пишет что не найден  За статью спасибо, очень полезно |  |  | | --- | | **0**  http://s5.ucoz.net/img/icon/thumbu_.png http://s5.ucoz.net/img/icon/thumbd_.png  **[3](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113" \l "ent82)**    **[ratboy](javascript://)**   (21.04.2012 17:40)     исправил |  |  | | --- | | **0**  http://s5.ucoz.net/img/icon/thumbu_.png http://s5.ucoz.net/img/icon/thumbd_.png  **[4](http://www.excelworld.ru/publ/vba/class_module/rabota_s_kollekcijami_klassov/42-1-0-113" \l "ent782)**    **[doctorraz](javascript://)**   (29.03.2015 10:32)     классная статья, многое стало понятно, но как всегда возник вопрос:  можно ли коллекции разных котиков не перебирая все коллекции в цикле, сгруппировать например по цвету,  т.е. на выходе получить  рыжых-5  белых-2  неважно как, средствами VBA Excel или чистого VBA | | |
|  | |
|  | |

Добавлять комментарии могут только зарегистрированные пользователи.